Université de Tiaret
Faculté MI
Département d'Informatique

Année universitaire : 2019/2020
Matière : Algorithmique et structure de données 1

Fiche de TD n° 2

« La structure alternative »

Exercice 1: Ecrire un algorithme (puis un programme en **langage C)** qui lit trois valeurs entières (A, B et C) au clavier et qui affiche la plus grande des trois valeurs, en utilisant:

1. **if – else** et une variable d'aide **Maximum**
2. **if - else if - ... – else** sans variable d'aide
3. les opérateurs conditionnels et une variable d'aide **Maximum**
4. les opérateurs conditionnels sans variable d'aide

Solution :

1)

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int A, B, C, max;
      scanf("%d%d%d",&A, &B, &C);
      if(A>B)
            max = A;
      else
            max = B;
      if(C>max)
            max = C;
      printf("le maximum entre A, B et C est : %d",max);
    return 0;
}
```

2)

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int A, B, C;
      scanf("%d%d%d",&A, &B, &C);
      if((A>B)&&(A>C))
            printf("le maximum entre A, B et C est : %d",A);
      else
            if((B>A)&&(B>C))
                  printf("le maximum entre A, B et C est :
%d",B);
            else
                  printf("le maximum entre A, B et C est :
%d",C);
    return 0;
}
```

3)

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int A, B, C, max;
      scanf("%d%d%d",&A, &B, &C);
      max = A>B?A:B;
      max = C>max?C:max;
      printf("le maximum entre A, B et C est : %d",max);
     return 0;
}
```

4)

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int A, B, C;
      scanf("%d%d%d",&A, &B, &C);
      printf("le maximum entre A, B et C est : %d",A>B&&A>C? A
: B>A&&B>C? B : C);
     return 0;
}
```

<u>Exercice 2:</u> Ecrire un algorithme (puis un programme en **langage C)** qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de leur produit et de leur somme sans faire la multiplication et sans faire l'addition.

Solution :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int A, B;
      scanf("%d%d",&A, &B);
      if ((a == 0) || (b == 0)) {
            printf("Le produit est nul\n");
      } else if (((a < 0) && (b < 0)) || ((a > 0) && (b > 0)))
{
            printf("Le produit est positif\n");
      } else {
            printf("Le produit est negatif\n");
      }
}
```

<u>Exercice 3</u>: Ecrire un algorithme (puis un programme en langage C) qui teste si un nombre entier est pair **sans la fonction modulo** «« *trouver un autre moyen de le résoudre* » ».Dans le cas où le chiffre est pair, vous afficherez "Le chiffre X est pair" et vous afficherez également "Le chiffre X est impair" si le chiffre est impair, où X sera le chiffre préalablement lu.

Solution :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
      int X,Y;
```

```
        scanf("%d",&X);
        Y = X/2;//Y prend le résultat de la division entière
        if(Y*2 == X)
                printf("%d est paire",X);
        else
                printf("%d est impaire",X);
    return 0;
}
```

Exercice 4: Ecrire un algorithme (puis un programme en **langage C)** qui calcule les solutions réelles d'une équation du second degré $ax^2+bx+c = 0$.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
        float a,b,c,x1,x2,delta;
        printf("Donner les valseurs de a, b et c");
        scanf("%f%f%f",&a,&b,&c);
        delta = pow(b,2)-4*a*c;
        printf("delta = %f\n",delta);
        if(delta >0)
        {
                x1 = (-b-sqrt(delta)) / (2*a);
                x2 = (-b+sqrt(delta)) / (2*a);
                printf("x1 = %f et x2 = %f",x1,x2);
        }else
                if(delta ==0)
                        {
                                x1 = x2 = -b/(2*a);
                                printf("x = %f",x1);
                        }
                else
                        printf("Il n'y a pas de solution r\éel");
    return 0;
}
```

Exercice 5: Ecrire un algorithme (puis un programme en **langage C)** qui lit trois valeurs entières (A, B et C) au clavier. Triez les valeurs A, B et C par échanges successifs de manière à obtenir : val (A) val(B) val(C). Afficher les trois valeurs.

Solution :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
        int A, B, C, temp;
        printf("Introduisez trois nombres entiers :");
        scanf("%d %d %d", &A, &B, &C);
        printf("\nAvant le tri : \tA = %d\tB = %d\tC = %d\n", A,
B, C);
        if (A < B) {
                temp = A;
                A = B;
                B = temp;
        }
        if (A < C) {
                temp = A;
                A = C;
```

```
            C = temp;
        }
        /* trier B et C */
        if (B < C) {
            temp = B;
            B = C;
            C = temp;
        }
        printf("Après le tri : \tA = %d\tB = %d\tC = %d\n", A, B,
C);
        return 0;
}
```

Exercice 6: Ecrire un algorithme (puis un programme en **langage C)** qui détermine si une année est bissextile ou non.
Si l'année A n'est pas divisible par 4, alors elle n'est pas bissextile Si A est divisible par 4, l'année est bissextile sauf si A est divisible par 100 et pas par 400.

Exemples :

– 1901 n'est pas bissextile car non divisible par 4
– 2004 est bissextile car divisible par 4 et pas par 100
– 2100 n'est pas bissextile car divisible par 4, divisible par 100 mais pas par 400
– 2000 est bissextile car divisible par 4, par 100 et par 400

Solution :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
        int annee;
        printf("Entrez une année: ");
        scanf("%d", &annee);
        if (annee % 4 == 0) {
            if (annee % 100 == 0) {
                if (annee % 400 == 0)
                    printf("%d est une année bissextile",
annee);
                else
                    printf("%d n\' est pas une année
bissextile", annee);
            } else
                printf("%d est une année bissextile", annee);
        } else
            printf("%d n\' est pas une année bissextile",
annee);
        return 0;
}
```

Exercice 7: Ecrire un algorithme (puis un programme en **langage C)** afficher la mention obtenue par un élève en fonction de la moyenne de ses notes. S'il a une moyenne strictement inférieure à 10, il est recalé. S'il a une moyenne entre 10 (inclus) et 12, il obtient la mention passable. S'il a une moyenne entre 12 (inclus) et 14, il obtient la mention assez bien. S'il a une moyenne entre 14 (inclus) et 16, il obtient la mention bien. S'il a une moyenne supérieure à 16 (inclus) il obtient la mention très bien.

Solution :

```
#include <stdio.h>
```

```
#include <stdlib.h>
int main() {
      int note;
      printf("introduiser une note:");
      scanf("%d", &note);
      if (note < 10) {
            printf("Recale\n");
      } else if (note < 12) {
            printf("Passable");
      } else if (note < 14) {
            printf("Assez bien\n");
      } else if (note < 16) {
            printf("Bien\n");
      } else {
            printf("Tres bien\n");
      }
      return 0;
}
```

Exercice 8: Ecrire un algorithme (puis un programme en **langage C)** permettant à l'utilisateur de rentrer 4 valeurs booléennes (0 pour faux, 1 pour vrai) indiquant si un plat est sucré ou pas, salé ou pas, chaud ou froid, cher ou pas. Le programme affichera l'une des reactions:

Un *Love:* si le plat est sucré, pas salé, froid et pas cher,
Un *Haha:* si le plat est salé, pas sucré, chaud,
Un *Wow*: si le plat est pas salé, pas sucré et froid,
Un *Sad*: si le plat est sucré, salé, froid,
Un *Angry*: sinon.
On utilisera des variables booléens sales, sucre, chaud et cher valant 0 ou 1